DOCUMENT RESUME

ED 119 526 FL 007 554

AUTHOR Burton-Hunter, Sarah K.
TITLE Etymology: A Computerized Model.
PUB DATE Apr 75
NOTE 48p.; Paper presented at International Conference of Computers in the Humanities (2nd, Los Angeles, California, April, 1975)

EDRS PRICE MF-$0.83 HC-$2.06 Plus Postage
DESCRIPTORS *Computational Linguistics; Computer Programs; *Diachronic Linguistics; *Etymology; *French; Language Patterns; *Latin; Models; Phonology; Suprasegmentals; Syllables; Vowels

ABSTRACT

Under the assumption that, with the exception of certain learned, retarded, and borrowed words, the bulk of any language undergoes sound changes that are regular over any given geographical area, over any given time span, and in any given sound environment, these sound changes have been reduced to logical terms and have been programmed to generate regularly derived cognate forms from a "proto, language. Classical Latin and Old French were chosen for the prototype, with an intermediate stage in Vulgar Latin. The computer has been programmed to recognize the whole of the sound environment: to recognize and place accents; to recognize a "syllable," a penult, antepenult, tonic and intertonic vowels, etc.; and to note qualities and quantities of vowels. Sound-change patterns have been programmed. The only data input is the Classical Latin word with vowels marked long or short; the program generates the changes the word has undergone and prints out the resulting forms in Vulgar Latin and Old French. The study has proven the feasibility of developing a program delineating regular sound-change patterns and generating etymological dictionaries of regularly derived cognate forms for a given proto-language, derivative languages and dialect variations. (Author/CLK)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. Nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

ED119526

Sarah K. Burton-Hunter

Etymology: A Computerized Model

Department of English

The University of Alabama

FL007554

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

Etymology: A Computerized Model

Despite the fact that the majority of sound changes within a given geographical area, within a given span of time, and within a given sound environment have long been considered regular and reduceable to definite patterns of change, and despite the fact that these changes have almost always been noted as variations of "If, then" statements, there has not yet been, to my knowledge, an attempt to program these patterns and statements into a computer-generated etymological dictionary of regularly derived cognate forms--a dictionary which could surpass in scope and accuracy any etymological dictionary yet produced, and one which offers the promise of being an important instructional tool. Such a dictionary would differentiate between regular formations and those forms which, for whatever reasons, did not follow regular phonological changes, and could note what the expected derivation would have been. The intrinsic value of the study lies not in the computerization of phonological changes occurring in the development of a Romance language, per se--though that is, in itself, a worthwhile accomplishment. The real value lies, rather, in the possibilities afforded by the expansion of

the program. At present, however, the program is still in its infancy: morphological and analogical variations to normal development have not been programmed; discriminations in the exact stages of development are not precise, with output printed arbitrarily as Classical Latin, Vulgar Latin, and Old French; the development is assumed to be that of the Francian dialect, though some exceptions may be found. The program as it stands traces possible word-histories without considering the effects of morphology, analogy, retardation or other factors which alter "regular" or expected sound-change patterns. This, to me, seems a sufficient beginning.

Many of my colleagues suggested that I tape and store the contents of the etymological dictionaries already in existence, those which had been produced by years of laborious and tedious tracing by hand, but the program which I envisioned was to serve as a model for a program which could, conceivably, generate a reconstructed Proto-IndoEuropean sound base as well as all the regularly derived cognate forms for all PIE languages and their dialects. Such a program would provide scholars with non-attested formations; with formations that have not survived or with variants which might not otherwise be recognized; with information dating retarded forms; with an aid for quickly spotting borrowed formations; with all the cognate forms

of any regular PIE derivative--and all these possibilities within the grasp of any interested party regardless of linguistic training. In these days of declining interest in languages, particularly in "non-relevant" languages, I see such a program as possibly serving future generations in much the same way as the Celtic monks served the continent of Europe upon their recall by Charlemagne.

Classical Latin and Old French were chosen for the prototype because of accessibility and attestability. Classical Latin is a well organized, well documented source, and Old French, also organized and documented, was, of the Romance languages, the farthest removed from Classical Latin and had undergone the greatest changes. The program prints out under the heading "Old French" formations of the Francian dialect of the early 12th Century and an intermediate stage labeled as "Vulgar Latin." While many formations at this intermediate stage are non-attested, the step is a useful one in many respects. The print-out also carries messages which identify the accented vowel; which note quality and quantity descriptions of the vowels; which carry notations indicating early or late loss of penult; and which will in the future cite orthographic variations, information which quite often is partially or totally absent from etymological dictionaries.

Because of the limitations of the number of symbols certain discriminations in vowels are lacking, such as in

bucina in which the pretonic u is labeled as VL u-closed. The program treats the symbol with regard to its pretonic position, but the label for the symbol lacks precision. The program itself makes many more discriminations than the labels and printed output indicate. The output does not make clear the differences in development between labra to levre and larga to large though the program does indeed discriminate between the effects of stop plus liquid and palatals preceded by consonants as indicated in the output by the resultant change in the accented vowel. Certain redundancies, such as labeling unaccented vowels in infinitive endings as 'open,' have proven convenient at this stage of the program, but should be refined when morphological considerations are added to the program.

The only data the machine receives is the Classical Latin word which is perceived as a string of characters. The vowels are differentiated as long or short. The string is analyzed for vowels and consonants. The first major step in the program is the placing of the accent and the syllabification of the word. Following Allen and Greenough's New Latin Grammar, vowel position is counted from the end of the word with the accent placed "on the penult if long or on the antepenult if the penult is short."[1] The factors determining the length of penult have also been programmed, as well as accentuation for words containing only one or

two syllables and some words of three syllables. Factors determining syllabification have also been programmed; and again, the analysis of the character string is the key. The complete Latin Accent subroutine is illustrated in Figure 1.

```
TLTH*TPFS.TLAT
      1              SUBROUTINE TLAT(WD,$)
      2              IMPLICIT INTEGER (A-Z)
      3              INCLUDE PARM,LIST
      4              INCLUDE LATIN,LIST
      5              INTEGER WD(1),WD2(RSZ),NPROC(7)
      6              DATA NPROC/3*' ','-NOT PROCESSED',0/
      7              INTEGER LMT1(LNCHR)/5*0,6*1,0,1,17*0/
      8              NLV=0
      9              CALL NBTW(' ',WD2,RSZ)
     10              CALL NBTW(0,LMES,3*RSZ)
     11              CALL NBTW(0,LVOW,20)
     12              CALL NBTW(0,LWD,RSZ)
     13              CALL SEP3(WD,WD2)
     14              DO 1 I=1,RSZ
     15              IF(WD2(I)),3
     16              DO 2 K=1,LNCHR
     17              IF(WD2(I)-LCHR(K)) 2,,2
     18              LWD(I)=K
     19              CALL MES(LMES(1,I),LMT1(K))
     20              IF(LVOWT(K)) 1,1
     21              NLV=NLV+1
     22              LVOW(1,NLV)=I
     23              GO TO 1
     24         2    CONTINUE
     25              CALL XFER(WD(1),NPROC,3)
     26              CALL NPRINT(6,NPROC)
     27              RETURN 2
     28         1    CONTINUE
     29         3    RETURN
     30              END


@PRT,S     LACC
```

FIGURE 1

LATIN ACCENT SUBROUTINE

The program notes the shift from quantity to quality in the vowel shift from Classical to Vulgar Latin as well as the raising and lowering of certain accented vowels.

Changes which unaccented vowels undergo in developing from Classical Latin to Vulgar Latin are largely dependent upon their position within the words. The unaccented vowels are, therefore, differentiated as initial, antepenultimate, penultimate, or final, and are treated accordingly, with resultant changes and losses programmed. (See Figure 2.) Messages are printed noting early and late loss of penult, information which is not generally included in etymological dictionaries, but which could be of some assistance to students of etymology. A portion of the subroutine which changes Classical Latin vowels to Vulgar Latin is included in Figure 2.

```
      C
      C     VOWELS
      C
215   POS=POS+1
      I2=IL+1
      I2=LWD(I2)
      IF(I.NE.2.OR.I.NE.3) GO TO 217 @ UNACC SHORT I+ VOWEL
      IF(LVOW(2,POS).EQ.1.OR.LVOWT(I2).NE.1) GO TO 217
      GO TO 218
217   IF(I.NE.3.OR.LVOWT(I2).NE.1) GO TO 220 @ I+LONG VOWEL
      IF(LVOW(2,POS+1).NE.1.OR.LVLS(I2).NE.1) GO TO 220
218   VWD(IV)=9
      CALL MES(VMES(1,IV),7)
      IL=IL+1
      I=LWD(IL)
      IV=IV+1
      GO TO 260
220   IF(WHERE-IL) 260,,260
      C
      C     PENULT --- LOOK FOR EARLY LOSS
      C
      I1=IL-1
      I1=LWD(I1)
      IF(I-3) 225,,225
      IF(LVOWT(I1).NE.0.OR.LVOWT(I2).NE.1) GO TO 225 @ I+VOWEL
      IL=IL+1
      I=LWD(IL)
      IO=IV-1
      VWD(IO)=9

      CALL MES(VMES(1,IV),7)
      GO TO 265
225   IF(LLIQ(I1)),,230  @LIQUID
      IF(I1-20) ,230, @ 'S'
      IF(DSTPS(I1)*DSTPS(I2)) ,,230
      IF(LSTPS(I1)*LSTPS(I2)) 240,240,230
230   CALL MES(VMES(1,IV),6)  @ EARLY LOSS
      GO TO 250
240   CALL MES(VMES(1,IV),5)  @ LATE LOSS
250   VWD(IV)=29
      GO TO 265
260   VWD(IV)=TCLVL(I)
      NVV=NVV+1
      VVOW(1,NVV)=IV
      I3=VWD(IV)
      CALL MES(VMES(1,IV),MCLVL(I3))
      PREVOW=LVOW(1,NVV)
      PREVOW=VWD(PREVOW)
      IF(PREVOW-29) 263,,263
      PLACE=NVV+1
      IF(LVOW(1,PLACE).NE.IL.OR.LVOW(2,PLACE).NE.1) GO TO 265
      VVOW(2,PLACE)=1
      CALL MES(VMES(1,IV),2)
```

FIGURE 2

VOWELS, CLASSICAL LATIN TO VULGAR LATIN

The effects of certain consonant groups and palatals upon the vowels are also incorporated into the routine. Thus far sound-change patterns affecting all vowels in words of one, two, and three syllables have been programmed. The major difficulty in programming the changes has been in ordering those changes which seem to have occurred almost simultaneously in the history of the development of the language.

Consonants, likewise, are scanned, and their positions are noted as initial, intervocalic, final, or clustered. Further, the consonants are identified as voiced or voiceless, with both mode and point of articulation noted. General rules are then programmed concerning the behavior or, for example, dental stops, stops plus liquids, or palatals. Behavior of specific sounds, such as the loss of aspirate h and peculiar changes such as -rs->-ss-, is also accounted for. The program provides for the insertion of glide consonants where needed in the Vulgar Latin to Old French section, and for unvoicing or loss of certain final consonants and occasional internal assimilations. In fact all the consonant changes have been incorporated into the program with the exception of the rather complex behavior of internal and final palatals and certain specific exceptions. Figure 3 illustrates the handling of certain intervocalic palatals, final consonants, and secondary consonant groups from Vulgar Latin to Old French.

```
285                GO TO 300
286          250   IF(J.NE.13.AND.J.NE.14) GO TO 252
287                IF(VVOWT(K1)-1) 251,
288                OWD(IO)=27 @ D,T
289                CALL MES(OMES(1,IO),13)
290                GO TO 295
291          251   IF(VVOWT(K2)-1) 260,
292                IF(K1.NE.26) GO TO 260 @ (T)H
293                OWD(IO)=10
294                IO=IO+1
295                IV=IV+1
296                GO TO 300
297          252   CONTINUE
298                IF(J.NE.12.AND.J.NE.28) GO TO 253 @ B, BIL. SP
299                OWD(IO)=13
300                GO TO 295
301          253   IF(VVOWT(K1)-1) 260,
302                IF(NVV.NE.AT+1) GO TO 254
303                ZZ=VVOW(1,NVV)
304                IF(VWD(ZZ).EQ.1.OR.VVOW(2,AT+1).EQ.1) GO TO 25
305                IV=IV+1
306                V1=V1+1
307                L1=L1+1
308                L2=IV+2
309                L3=L2+1
310                K1=K2
311                K2=K3
312          254   IF(K2) ,260,
313                IF(VWD(IV)-24) 257,,257 @ G
314                IF(K1-1) 260,,260 @ A
315                DO 255 I1=1,4
316                IF(H1-IOTB2(1,I1))     ,300, @ O,U+ G+ A
317          255   CONTINUE
318                OWD(IO)=6
319                GO TO 295
320          257   IF(J-20) 258,,258  @ N+ E,I
321                IF(K1.NE.2.AND.K1.NE.3)  GO TO 260
322                OWD(IO)=17
323                OWD(V1)=6
324                V1=V1+1
325                OWD(V1)=4
326                IO=IO+2
327                IV=IV+1
328                GO TO 295
329          258   IF(J-23) 260,,260 @ C+ E,I
330                IF(K1.NE.2.AND.K1.NE.6.AND.K1.NE.3) GO TO 290
331                OWD(IO)=6
332                CALL MES(OMES(1,IO),7)
333                IO=IO+1
334                OWD(IO)=14
335                GO TO 295
336          C     FINAL CONSONANTS
337          260   L5=L2+3
338                L4=L2+2
339                K5=VWD(L5)
340                K4=VWD(L4)
341                IF(K5) 290,,290
```


FIGURE 3

CONSONANT CHANGE, VULGAR LATIN TO OLD FRENCH

Once the Classical Latin string has been analyzed and altered to its Vulgar Latin form, the Vulgar Latin string is similarly scanned, analyzed, and altered to the resultant Old French string. The Classical Latin word and both its consequent strings with changes, losses, or insertions, as required, are then printed in three columns; pertinent messages are noted next to the appropriate character in the separate columns. Such a format seems to me to offer useful information in the most accessible manner, and will allow for ease of comparison when future languages are added, as conceptualized in Figure 4.

CLASSICAL LATIN
FACTUM
F
A - Accented
C
T
U
M

VULGAR LATIN, WESTERN
FAITO
F
A - Accented
I - Yod
T
O - Long

OLD FRENCH
FAIT
F
A
I
T

OLD PROVENÇAL
FACH
F
A
C
H

OLD SPANISH, 12th CENTURY
FECHO
F
E
C
H
O

OLD SPANISH, 15th CENTURY
HECHO
H
E
C
H
O

OLD PORTUGUESE
FEITO
F
E
I
T
O

VULGAR LATIN, EASTERN
FATTO
F
A - Accented
T
T
O - Long

OLD ITALIAN
FATTO
F
A
T
T
O

FIGURE 4

CONCEPTUALIZED EXPANSION OF ROMANCE ETYMOLOGY

To date the program incorporates only phonological changes. When and if funding becomes available, I plan to incorporate a morphological subroutine. I am hopeful that the computer can be instructed to differentiate between nominal and verbal forms, to decline nouns, and to conjugate verbs. At the same time such a program should take into account exceptions to regular phonological change due to analogy and reduction in the number of declensions and conjugations, as well as other morphological exceptions to regular phonological change.

The purpose of this paper has been to illustrate the relative ease with which the computer can be adapted to phonological and etymological research, not to dwell on details. I do believe that utilization of the computer will make possible the compilation of dictionaries of more cognate forms of more languages than lifetimes of work could have accomplished previously. While I have yet to attempt what is perhaps the most controversial element of etymological studies, reconstruction of non-attested languages, I further believe that use of the computer will certainly aid in the compilation of factors of common heritage among derivative languages, common relationships which are, of course, the key to reconstructive efforts.

A noted linguist has asked me what I have accomplished that had not been previously accomplished with the applica-

tion of a lot of elbow grease. The question seems to me rhetorical. By eliminating for future scholars the elbow grease now required for etymological studies, particularly in comparative fields, work which could have been a lifetime's undertaking might now be undertaken as a sabbatical project, or as a year's work for a graduate seminar, or as a brief funded research project. Each language history which is programmed, assuming some form of standardization, can be added to the files available to scholars, and each addition increases greatly the probability that the time will come when the entire family of Indo-European languages will be instantly available for comparative, anthropological, historical, and etymological studies.

Aside from the ease and convenience of computer storage of such a wealth of material, a veritable treasure of a word-hoard, there is yet another aspect to be considered, and that is the statistical analysis of language change which would be made possible. Long standing questions such as the extent to which language change is regular could at last be answered. Of course, I speak of future possibilities. I had planned the first expansion of the Romance model to be the generation of the koiné of Old French, and the addition of other Romance languages, but my limitations have been the nemeses of all researchers, too little time, too little knowledge, and too little money. I have, therefore, appended to this report a substantial portion of my

program in the hope that it may prove valuable to other researchers.[2]

NOTES

[1]Allen and Greenough's New Latin Grammar, eds. J. B. Greenough, G. L. Kittredge, A. A. Howard, Benjamin L. D'ooge (Boston: Ginn and Company, 1888, rev. and rptd. 1903, 1916, 1931), p. 7.

[2]Thanks are due Professor Jan A. Nelson and Professor Carl Seebeck of the University of Alabama and are due the two programmers without whom the project would have been an impossibility, Harry Lee Pendergrass and Thomas L. Thornton, members of a Computer Honors seminar. Their contribution has been a crucial factor in the success of the program. Finally, thanks are due the Graduate Council of the University of Alabama, who by twice awarding me a Graduate Council Research Fellowship, granted me time and freedom to work on my project.

Appendix I

Sample data output

CLASSICAL LATIN

DENTEM

D
E- ACCENTED
N
T
E
M

VULGAR LATIN

DENTE

D
E- OPEN
- ACCENTED
N
T
E- OPEN

OLD FRENCH

DENT

D
E- ACCENTED
N
T

LABRA

L
A- ACCENTED
B
R
A

LABRA

L
A- ACCENTED
B
R
A

LEVRE

L
E- ACCENTED
V
R
E

LARGA

L
A- ACCENTED
R
G
A

LARGA

L
A- ACCENTED
R
G
A

LARGE

L
A- ACCENTED
R
G
E

CALERE

C
A
L
E- LONG
 - ACCENTED
R
E

CALERE

C
A
L
E- CLOSE
 - ACCENTED
R
E- OPEN

CHALEIR

C
H
A
L
E- ACCENTED
I
R

LOCARE

L
O
C
A- LONG
 - ACCENTED
R
E

LOGARE

L
O- OPEN
G
A- ACCENTED
R
E- OPEN

LOER

L
O
E- ACCENTED
R

BUCINA

B
U- LONG
C
I- LONG
 - ACCENTED
N
A

BUCINA

B
U- CLOSE
C
I- CLOSE
 - ACCENTED
N
A

BUISINE

B
U
I- YOD
S
I- ACCENTED
N
E

FACTUM

F
A- ACCENTED
C
T
U
M

FAITO

F
A- ACCENTED
I- YOD
T
O- CLOSE

FAIT

F
A- ACCENTED
I
T

GAMBA

G
A- ACCENTED
M
B
A

GAMBA

G
A- ACCENTED
M
B
A

JAMBE

J
A- ACCENTED
M
B
E

GENTEM

G
E- ACCENTED
N
T
E
M

IENTE

I- YOD
E- OPEN
 - ACCENTED
N
T
E- OPEN

GENT

G
E- ACCENTED
N
T

FIRMUM

F
I- ACCENTED
R
M
U
M

FERMO

F
E- CLOSE
- ACCENTED
R
M
O- CLOSE

FER

F
E- ACCENTED
R

FORUM

F
O- ACCENTED
R
U
M

FORO

F
O- OPEN
- ACCENTED
R
O- CLOSE

FUER

F
U- ACCENTED
E
R

CLASSICAL LATIN

FRENUM

F
R
E- LONG
- ACCENTED
N
U
M

VULGAR LATIN

FRENO

F
R
E- CLOSE
- ACCENTED
N
O- CLOSE

OLD FRENCH

FREIN

F
R
E- ACCENTED
I
N

CLASSICAL LATIN	VULGAR LATIN	OLD FRENCH
MATURUM	MATURO	MERUR
M	M	M
A	A	E
T	T	R- THORN *
U- LONG	U- CLOSE	U- ACCENTED
- ACCENTED	- ACCENTED	R
R	R	
U	O- CLOSE	
M		
USARE	USARE	USER
U- LONG	U- CLOSE	U
S	S	S
A- LONG	A- ACCENTED	E- ACCENTED
- ACCENTED	R	R
R	E- OPEN	
E		
VIVERE	VIV°RE	VIVRE
V	V	V
I- LONG	I- CLOSE	I- ACCENTED
- ACCENTED	- ACCENTED	V
V	V	R
E	°- LATE LOSS	E
R	R	
E	E- OPEN	

Appendix II

Major subroutines

```
1*TPFS.PROCS
  1       PARM PROC
  2            PARAMETER RSZ=18
  3            PARAMETER NMES=14
  4            PARAMETER LNCHR=30
  5            PARAMETER PU=33
  6        END
  7       LATIN PROC
  8            COMMON/LBLK/LWD(RSZ),NLV,LVOW(2,10),LMES(3,RSZ)
  9            COMMON/LBLK2/LCHR(LNCHR),LVOWT(LNCHR),LVLS(LNCHR),LPAL(LNC
 10           *DSTPS(LNCHR),LSTPS(LNCHR),LLIQ(LNCHR),LSTOP(LNCHR)
 11            DATA LCHR/'A','E','I','O','U','1','2','3','4','5','6','7',
 12           $'B','T','D','F','V','S','Z','M','N','R','L','C','G','X','H
 13            DATA LVOWT/13*1,17*0/ @ 1=VOWEL, 0=NOT
 14            DATA LVLS/5*0,6*1,0,1,17*-1/ @ 1=LONG, 0=SHORT
 15            DATA LPAL/0,1,1,3*0,1,1,22*0/ @ 1=PAL., 0=NOT
 16            DATA DSTPS/15*0,1,1,13*0/ @ 1=DENT. STP., 0=NOT
 17            DATA LSTPS/13*0,1,1,15*0/ @1=LAB. STP., 0=NOT
 18            DATA LLIQ/23*0,1,1,5*0/ @1=LIQUID, 0=NOT
 19            DATA LSTOP/13*0,4*1,13*0/ @ 1=STOP, 0=NOT
 20       END
 21       MESAG PROC
 22            INTEGER PMESS(4,NMES)
 23            DATA (PMESS(I,1),I=1,4)/'LONG',3*' '/
 24            DATA (PMESS(I,2),I=1,4)/'ACCENTED',2*' '/
 25            DATA (PMESS(I,3),I=1,4)/'OPEN',3*' '/
 26            DATA (PMESS(I,4),I=1,4)/'CLOSE',3*' '/
 27            DATA (PMESS(I,5),I=1,4)/'LATE LOSS',2*' '/
 28            DATA (PMESS(I,6),I=1,4)/'EARLY LOSS',2*' '/
 29            DATA (PMESS(I,7),I=1,4)/'YOD',3*' '/
 30            DATA (PMESS(I,8),I=1,4)/'WAU',3*' '/
 31            DATA (PMESS(I,9),I=1,4)/'BILABIAL SPIRANT',' '/
 32            DATA (PMESS(I,10),I=1,4)/'OR C ...MSS VARY',' '/
 33            DATA (PMESS(I,11),I=1,4)/'OR G ...MSS VARY',' '/
 34            DATA (PMESS(I,12),I=1,4) /'(DENTAL STOP VOICES OUT)'/
 35            DATA (PMESS(I,13),I=1,4) /'THORN',3*' '/
 36            DATA (PMESS(I,14),I=1,4)/'PROSTHETIC',2*' '/
 37       END
 38       VULGR PROC
 39            INTEGER MCLVL(29)/0,4*4,3*3,7,8,19*0/ @MESSAGE TAGS
 40            COMMON/VBLK/VWD(RSZ),NVV,VVOW(2,10),VCHR(29),VMES(3,RSZ)
 41           $,VVOWT(29),VSTPS(29),VPAL(29),VLIQ(29),VNAS(29)
 42            DATA VCHR/'A','E','I','O','U','E','O','U','I','U','P','B',
 43           $'D','F','V','S','Z','M','N','R','L','C','G','X','H','Q','S
 44            DATA VVOWT/8*1,21*0/ @ 1=VOWEL, 0=NOT
 45            DATA VSTPS/10*0,4*1,15*0/ @ 1=STOP, 0=NOT
 46            DATA VPAL/0,1,1,0,0,1,2*0,1,20*0/ @ 1=PAL.,0=NOT
 47            DATA VLIQ /20*0,2*1,7*0/ @ 1=LIQUID, 0=NOT
 48            DATA VNAS /18*0,2*1,9*0/ @ 1=NASAL, 0=NOT
 49       END
 50       OFRNC PROC
 51            COMMON/OBLK/OWD(RSZ),NOV,OVOW(2,10),OCHR(27),OMES(3,RSZ),O
 52            DATA OCHR/'A','E','I','O','U','I','U','P','B','T','D','F',
 53           *'S','Z','M','N','R','L','C','G','X','H','Q','J','S','E'/
 54       END
```

```
1*TPFS.INLAT
  1             IMPLICIT INTEGER (A-Z)
  2             INCLUDE PARM,LIST
  3             INTEGER CD(14)
  4             CALL NPRINT(PU, '                    EVOLUTION OF LANGUAGES  PART 1--
  5            * ROMANCE ETYMOLOGY . ')
  6             CALL NPRTCN(PU,'L,3 .')
  7             CALL NPRINT(PU,'                      SARAH K. BURTON- HUNTER, PRINC
  8            *IPLE INVESTIGATOR . ')
  9             CALL NPRTCN(PU,'L,5 .')
 10             CALL NPRINT(PU,'                       T THORNTON, H PENDERGRASS, PR
 11            *OGRAMMING ASSTS . ')
 12             CALL NPRTCN(PU,'L,7 .')
 13             CALL NPRINT(PU,'                     UNIVERSITY OF ALABAMA COMPUTER
 14            *CENTER FACILITIES . ')
 15             CALLNPRINT (PU,'                     ADDRESS INQUIRIES TO  . ')
 16             CALL NPRINT(PU,'                            P.O. BOX 866, UNIVERSITY
 17            *, AL. 35486 . ')
 18             CALL NPRTCN(PU,'H,X,2,    CLASSICAL LATIN                    VULGAR
 19            *LATIN                  OLD FRENCH . ')
 20             CALL NPRINT(PU,0,55)
 21             CALL NPRTCN(PU,'L,2 .')
 22             CALL NPRINT(PU,0,60)
 23       100   CALL CREAD(CD,$200)
 24             DO 110 I=1,10,3
 25             IF(CD(I).EQ.' ') GO TO 110
 26             CALL TLAT(CD(I),$110)
 27             CALL LACC
 28             CALL LTOV
 29             CALL VTOF
 30             CALL PLAT
 31             CALL PVUL
 32             CALL POFR
 33             CALL PPG
 34       110   CONTINUE
 35             GO TO 100
 36       200   CONTINUE
 37             END


*,S      TLAT
```

```
 57              IV=2
 58              IF(LWD(3).NE.10.OR.LVOW(2,2).NE.1) GO TO 140
 59              I2=LWD(I2)
 60              IF(LVOWT(I2)) ,200, @ 8+CONS, ACC LONG U
 61        140   VWD(IL)=5
 62        150   IV=IV+1
 63              I=LWD(1)
 64              VWD(1)=TCLVL(I)
 65              N1=VWD(1)
 66              CALL MES(VMES(1,1),MCLVL(N1))
 67              IF(LVOW(1,1).NE.1) GO TO 200
 68              POS=1
 69              NVV=1
 70              VVOW(1,NVV)=1
 71              IF(LVOW(2,1).NE.1) GO TO 200
 72              VVOW(2,1)=1
 73              CALL MES(VMES(1,1),2)
 74        200   WHERE=0          @ POS OF PENULT
 75              IF(NLV-2) 210,
 76              IF(LVOW(2,NLV-1)),,210   @ACCENTED
 77              WHERE=LVOW(1,NLV-1)
 78        210   IF(LWD(IL)),300
 79              I=LWD(IL)
 80              IF(LVOWT(I)),,215
 81              VWD(IV)=TCLVL(I)
 82              IL=IL+1
 83              IV=IV+1
 84              GO TO 210
 85        C
 86        C     VOWELS
 87        C
 88        215   POS=POS+1
 89              I2=IL+1
 90              I2=LWD(I2)
 91              IF(I.NE.2.OR.I.NE.3) GO TO 217 @ UNACC SHORT I+ VOWEL
 92              IF(LVOW(2,POS).EQ.1.OR.LVOWT(I2).NE.1) GO TO 217
 93              GO TO 218
 94        217   IF(I.NE.3.OR.LVOWT(I2).NE.1) GO TO 220 @ I+LONG VOWEL
 95              IF(LVOW(2,POS+1).NE.1.OR.LVLS(I2).NE.1) GO TO 220
 96        218   VWD(IV)=9
 97              CALL MES(VMES(1,IV),7)
 98              IL=IL+1
 99              I=LWD(IL)
100              IV=IV+1
101              GO TO 260
102        220   IF(WHERE-IL) 260,,260
103        C
104        C     PENULT --- LOOK FOR EARLY LOSS
105        C
106              I1=IL-1
107              I1=LWD(I1)
108              IF(I-3) 225,,225
109              IF(LVOWT(I1).NE.0.OR.LVOWT(I2).NE.1) GO TO 225 @ I+VOWEL
110              IL=IL+1
111              I=LWD(IL)
112              IO=IV-1
113              VWD(IO)=9
```

```
114             CALL MES(VMES(1,IV),7)
115             GO TO 265
116       225   IF(LLIQ(I1)),,230  @LIQUID
117             IF(I1-20) ,230, @ 'S'
118             IF(DSTPS(I1)*DSTPS(I2)) ,,230
119             IF(LSTPS(I1)*LSTPS(I2)) 240,240,230
120       230   CALL MES(VMES(1,IV),6)  @ EARLY LOSS
121             GO TO 250
122       240   CALL MES(VMES(1,IV),5)  @ LATE LOSS
123       250   VWD(IV)=29
124             GO TO 265
125       260   VWD(IV)=TCLVL(I)
126             NVV=NVV+1
127             VVOW(1,NVV)=IV
128             I3=VWD(IV)
129             CALL MES(VMES(1,IV),MCLVL(I3))
130             PREVOW=LVOW(1,NVV)
131             PREVOW=VWD(PREVOW)
132             IF(PREVOW-29) 263,,263
133             PLACE=NVV+1
134             IF(LVOW(1,PLACE).NE.IL.OR.LVOW(2,PLACE).NE.1) GO TO 265
135             VVOW(2,PLACE)=1
136             CALL MES(VMES(1,IV),2)
137             GO TO 265
138       263   IF(LVOW(1,NVV).NE.IL.OR.LVOW(2,NVV).NE.1) GO TO 265
139             VVOW(2,NVV)=1
140             CALL MES(VMES(1,IV),2)
141       265   IV=IV+1
142             IL=IL+1
143             I1=IL+1
144             IF(LWD(I1)) 210,210,
145       270   I1=LWD(I1)
146             IF(LVOWT(I1)) 280,280
147             I=LWD(IL)
148             IF(VPAL(I1)) 275,,275 @ G+ A,0,U
149             IF(I-27) ,275,
150             DO 275 I2=1,5
151             IF(I -IVCTB1(1,I2)) 275,,275
152             VWD(IV)=IVCTB1(2,I2)
153             CALL MES(VMES(1,IV),IVCTB1(3,I2))
154             IV=IV+1
155             IL=IL+1
156             GO TO 210
157       275   CONTINUE
158             IF(I-26) 210,,210 @ C+A,0,U
159             IF(LPAL(I1)) 210,,210
160             VWD(IV)=24
161             IV=IV+1
162             IL=IL+1
163             GO TO 210
164       280   I1=IL+2
165             IF(LWD(I1)) 210,210,
166             I1=LWD(I1)
167             IF(LVOWT(I1)) 210,210,
168             I3=LWD(IL)
169             I4=LWD(IL+1)
170             DO 290 I2=1,7
```

```
      171               IF(IVCTB2(1,I2).NE.I3.OR.IVCTB2(2,I2).NE.I4) GO TO 290
      172               VWD(IV)=IVCTB2(3,I2)
      173               VWD(IV+1)=IVCTB2(4,I2)
      174               CALL MES(VMES(1,IV),IVCTB2(5,I2))
      175               IV=IV+2
      176               IL=IL+2
      177         290   CONTINUE
      178               GO TO 210
      179         300   IF(LWD(IL-1).NE.22.AND.LWD(IL-1).NE.23) RETURN
      180               IF (NLV.EQ.1) RETURN
      181               VWD(IV-1)=0
      182               RETURN
      183               END


&PRT,S      VTOF
```

```
TLTH*TPFS.VTOF
     1              SUBROUTINE VTOF
     2              IMPLICIT INTEGER(A-Z)
     3              INCLUDE PARM,LIST
     4              INCLUDE VULGR,LIST
     5              INCLUDE OFRNC,LIST
     6              INTEGER IOTB1(4,8)/1,25,10,11,
     7             *4,25,10,0,
     8             *7,25,10,0,
     9             *5,25,0,0,
    10             *8,25,0,0,
    11             *2,21,0,0,
    12             *6,21,0,0,
    13             *3,21,0,0/
    14              INTEGER IOTB2(1,4)/4,5,7,8/
    15              INTEGER IOTB3(3,8)/12,12,0,
    16             *13,0,12,
    17             *14,0,12,
    18             *16,17,0,
    19             *19,17,0,
    20             *20,17,0,
    21             *24,6,0,
    22             *28,12,0/
    23              INTEGER IVLOF(29)/1,2,3,4,5,2,4,5,6,7,8,9,10,11,12,13,14,15
    24             *                    18,19,20,21,22,23,24,13,0/
    25              LOGICAL INIT/.TRUE./
    26              LOGICAL FREE/.TRUE./
    27              AT=0 @ VOWEL POSITION
    28              NOV=0
    29              CALL NBTW(0,OWD,RSZ)
    30              CALL NBTW(0,OMES,3*RSZ)
    31              CALL NBTW(0,OVOW,20)
    32              IO=1
    33              IV=2
    34        C     INITIAL CONSONANTS
    35              IF(VWD(2)-1) 105,,105
    36              IF(VWD(1)-23) 100,,100 @C+ A
    37              OWD(IO)=20
    38              OWD(IV)=23
    39              IO=IO+2
    40              V1=IV+1
    41              IF(VWD(V1)-5) 70,,70 @ C+ AU(8)
    42              OWD(IO)=4
    43              AT=AT+1
    44              GO TO 75
    45        70    IF(VWD(V1).NE.22) GO TO 245
    46              OWD(IO)=1
    47              AT=1
    48        75    INIT=.NOT.INIT
    49              IF(VVOW(2,1).NE.2) GO TO 80
    50              CALL MES(OMES(1,IV),2)
    51        80    GO TO 295
    52        100   IF(VWD(1)-24) 105,,105 @G+ A
    53              OWD(IO)=25
    54              IO=IO+1
    55              GO TO 245
    56        105   IF(VWD(1)-9) 120,,120 @YOD+ VOWEL
```

```
57              I=VWD(IV)
58               INIT=.NOT.INIT
59              DO 110 I1=1,8 @ G
60              IF(I-IOTB1(1,I1)) 110,,110
61              OWD(IO)=IOTB1(2,I1)
62              IO=IO+1
63              GO TO 245
64        110   CONTINUE
65              GO TO 290
66        120   IF(VWD(IV)-5) 135,,135
67              I=VWD(IV+1)
68              IF(VWD(1).EQ.27) MO=3 @QU
69              IF(VWD(1)-24) 135,,130 @ GU
70        130   CONTINUE
71              DO 135 I1=1,8
72              IF(I-IOTB1(1,I1)) 135,,135
73              K=VWD(1)
74              J=VWD(IV)
75              OWD(IO)=TVLOF(K)
76              OWD(IV)=TVLOF(J)
77              AT=1
78              IO=IO+1
79              CALL MES(OMES(1,IV),IOTB1(MO,I1))
80              GO TO 295
81        135   IV=1
82              GO TO 245
83        C
84        C     VOWELS
85        140   AT=AT+1
86              IF(K1) ,145,
87              IF(K2) ,144,
88              IF(VSTPS(K1).EQ.1.AND.VLIQ(K2).EQ.1) GO TO 145
89              IF(VVOWT(K1).EQ.0.AND.VVOWT(K2).EQ.0) FREE=.NOT.FREE
90        144   IF(VWD(IV)-4) 145,,145 @ O+NASAL
91              IF(VNAS(K1)) 145,,145
92              IF(FREE) FREE=.NOT.FREE
93        145   IF(VVOW(2,AT)) ,,180 @ NOT ACCENTED
94              IF(.NOT.FREE) GO TO 290
95        C     FREE VOWELS
96              IF(K1) ,177,
97              IF(.NOT.INIT) GO TO 177
98              INIT=.NOT.INIT
99        C     INITIAL VOWELS
100             IF(VWD(IV).NE.2.AND.VWD(IV).NE.6) GO TO 148 @ E
101             OWD(IO)=2
102             IV=IV+1
103             IO=IO+1
104             IF(K1-9) 245,,245
105             OWD(IO)=3
106             AT=AT+1
107             GO TO 295
108       148   IF(VWD(IV)-3) 150,,150 @ I
109             OWD(IO)=3
110             IF(K1-9) 295,,295
111             AT=AT+1
112             IV=IV+1
113             GO TO 295
```

```
114       150   IF(VWD(IV)-1) 165,,165 @ A
115             IF(K1.NE.1.OR .K2.NE.22) GO TO 155
116             AT=AT+1
117             GO TO 300
118       155   IF(K1.NE.5.AND.K1.NE.8) GO TO 158 @ A+U
119             OWD(IO)=4
120             IV=IV+1
121             AT=AT+1
122             IF(K2-9) 295,,295 @ AU+YOD
123             OWD(V1)=3
124             IF(VVOW(2,AT+1).EQ.2) CALL MES(OMES(1,V1),2)
125             IV=IV+2
126             IO=IO+2
127             AT=AT+1
128             GO TO 245
129       158   IF(VVOWT(K1)) 160,,160 @ HIATUS A
130             IF(VVOW(2,AT+1)-1) 160,
131             IF(K2.NE.5.AND.K2.NE.8) GO TO 160
132             OWD(IO)=2
133             OWD(V1)=5
134             AT=AT+1
135             IV=IV+3
136             IO=IO+2
137             GO TO 245
138       160   IF(H1) ,290,
139             IF(H1.EQ.9.OR.H1.EQ.23.OR.H1.EQ.24) GO TO 177
140             OWD(IO)=1
141             IF(K1-9) 295,,295
142             OWD(V1)=3
143             AT=AT+1
144             IV=IV+1
145             IO=IO+1
146             GO TO 295
147       165   IF(VWD(IV)-7) 175,,175 @ OPEN O
148             IF(K1.NE.11.OR.K1.NE.22) GO TO 167
149             IF(K2-9) 170,,170
150             OWD(IO)=4
151             OWD(V1)=5
152             AT=AT+1
153             IV=IV+2
154             IO=IO+1
155             GO TO 295
156       167   IF(K1-9) 170,,170
157             AT=AT+1
158             OWD(IO)=5
159             OWD(V1)=3
160             IV=IV+1
161             IO=IO+1
162             GO TO 295
163       170   IF(H1) ,290,
164             IF(VNAS(H1)-1) 290,,290
165             OWD(IO)=4
166             OWD(V1)=5
167             IV=IV+1
168             IO=IO+2
169             GO TO 245
170       175   IF(VWD(IV)-4) 290,,290 @ CLOSED O
```

```
171             OWD(IO)=4
172             OWD(V1)=5
173             IV=IV+1
174             IO=IO+2
175             IF(K1-9) 245,,245
176             OWD(V1)=3
177             AT=AT+1
178             GO TO 245
179       C     INTERTONIC VOWELS
180       177   IF(J.NE.1) GO TO 178
181             OWD(IO)=2
182             GO TO 295
183       178   IF(K1) ,310,
184             GO TO 290
185       C     ACCENTED VOWELS
186       180   IF(INIT) INIT=.NOT.INIT
187             IF(.NOT.FREE) GO TO 220
188             IF(J-1) 185,,185 @ A
189             IF(H1),183,
190             IF(VPAL(H1)-1) ,184,  @ PAL+A
191       183   IF(K1) ,184,
192             IF(VNAS(K1)-1) 184,,184 @ FF NASAL
193             OWD(IO)=1
194             CALL MES(OMES(1,IO),2)
195             OWD(V1)=6
196             IO=IO+1
197             GO TO 295
198       184   OWD(IO)=2
199             CALL MES(OMES(1,IO),2)
200             GO TO 295
201       185   IF(J-2) 200,,200 @ CLOSED E
202             DO 190 I=1,18
203             IF(VWD(I)) 190,,190
204             Q=I-1
205             IF(VWD(Q).NE.3) GO TO 190
206             OWD(IO)=3
207             AT=AT+1
208             CALL MES(OMES(1,IO),2)
209             GO TO 295
210       190   CONTINUE
211             CALL MES(OMES(1,IO),2)
212             IF(VPAL(H1)-1) ,195
213             OWD(IO)=2
214             OWD(V1)=3
215             IO=IO+1
216             GO TO 295
217       195   OWD(IO)=3
218             GO TO 295
219       200   IF(J-6) 220,,220 @   OPEN E
220             IF(K1.EQ.0.AND.H1.EQ.3) GO TO 300
221             OWD(IO)=3
222             OWD(V1)=2
223             CALL MES(OMES(1,V1),2)
224             IO=IO+1
225             GO TO 295
226       C     ACCENTED VOWELS-- FREE OR CHECKED
227       220   IF(VWD(IV)-7) 230,,230
```

```
228             OWD(IO)=5
229             IF(VWD(L1)-9) 225,,225 @   OPEN O+ YOD
230             IV=IV+1
231             AT=AT+1
232             IO=IO+1
233             OWD(V1)=3
234             CALL MES(OMES(1,IO),2)
235             GO TO 295
236       230   IF(VWD(IV)-1) 235,,235 @ AU
237             IF(VWD(L1)-5) 240,,240
238             OWD(IO)=4
239             IV=IV+1
240             AT=AT+1
241             CALL MES(OMES(1,IO),2)
242             GO TO 295
243       225   IF(.NOT.FREE) GO TO 240
244             OWD(V1)=2
245             CALL MES(OMES(1,IO),2)
246             IO=IO+1
247             GO TO 295
248       235   IF(VWD(IV)-4) 240,,240 @ CLOSED O
249             IF(K1+K3) ,310,
250             OWD(IO)=4
251             CALL MES(OMES(1,IO),2)
252             IF(VNAS(K1)-1) ,295,
253             IO=IO+1
254             OWD(IO)=5
255             GO TO 295
256       240   CALL MES(OMES(1,IO),2)
257             GO TO 290
258       245   J=VWD(IV)
259             IF(J) 310,310,
260             L1=IV+1
261             L2=IV+2
262             L3=L2+1
263             V1=IO+1
264             IF(IV-1) ,247,
265             H1=VWD(IV-1)
266       247   IF(IV.EQ.1)     H1   =O
267             K1=VWD(L1)
268             K2=VWD(L2)
269             K3=VWD(L3)
270             IF(.NOT.FREE) FREE=.NOT.FREE
271             IF(VVOWT(J)) ,,140
272             IF(H1) ,260,
273             IF(VVOWT(H1)-1) 260,
274       C     INTERVOCALIC
275       C     CONSONANTS
276       C
277             IF(K1) ,260,
278             IF(J-K1) 250,,250
279             IF(K2) ,300,
280             IF(K3.NE.O.OR .VVOWT(K2).NE.1)  GO TO 248
281             VWD(L2)=O
282             GO TO 300
283       248   CONTINUE
284             IF(J.EQ.17.OR.J.EQ.21.OR.J.EQ.22) GO TO 290
```

```
285             GO TO 300
286       250   IF(J.NE.13.AND.J.NE.14) GO TO 252
287             IF(VVOWT(K1)-1) 251,
288             OWD(IO)=27 @ D,T
289             CALL MES(OMES(1,IO),13)
290             GO TO 295
291       251   IF(VVOWT(K2)-1) 260,
292             IF(K1.NE.26) GO TO 260 @ (T)H
293             OWD(IO)=10
294             IO=IO+1
295             IV=IV+1
296             GO TO 300
297       252   CONTINUE
298             IF(J.NE.12.AND.J.NE.28) GO TO 253 @ B, BIL. SPIRANT
299             OWD(IO)=13
300             GO TO 295
301       253   IF(VVOWT(K1)-1) 260,
302             IF(NVV.NE.AT+1) GO TO 254
303             ZZ=VVOW(1,NVV)
304             IF(VWD(ZZ).EQ.1.OR.VVOW(2,AT+1).EQ.1) GO TO 254
305             IV=IV+1
306             V1=V1+1
307             L1=L1+1
308             L2=IV+2
309             L3=L2+1
310             K1=K2
311             K2=K3
312       254   IF(K2) ,260,
313             IF(VWD(IV)-24) 257,,257 @ G
314             IF(K1-1) 260,,260 @ A
315             DO 255 I1=1,4
316             IF(H1-IOTB2(1,I1))    ,300, @ O,U+ G+ A
317       255   CONTINUE
318             OWD(IO)=6
319             GO TO 295
320       257   IF(J-20) 258,,258  @ N+ E,I
321             IF(K1.NE.2.AND.K1.NE.3)  GO TO 260
322             OWD(IO)=17
323             OWD(V1)=6
324             V1=V1+1
325             OWD(V1)=4
326             IO=IO+2
327             IV=IV+1
328             GO TO 295
329       258   IF(J-23) 260,,260 @ C+ E,I
330             IF(K1.NE.2.AND.K1.NE.6.AND.K1.NE.3) GO TO 290
331             OWD(IO)=6
332             CALL MES(OMES(1,IO),7)
333             IO=IO+1
334             OWD(IO)=14
335             GO TO 295
336       C     FINAL CONSONANTS
337       260   L5=L2+3
338             L4=L2+2
339             K5=VWD(L5)
340             K4=VWD(L4)
341             IF(K5) 290,,290
```

```
342             IF(K4-10) ,,290
343             IF(K1-29) 262,,262 @SECOND. CONS. GP.
344             IF(VVOWT(K2)) 265,,265
345             OWD(IO)=TVLOF(J)
346             IO=IO+1
347             IF(VMES(1,K1)-6) ,300,
348             OWD(IO)=TVLOF(K2)
349             IF(K3) ,261,
350             IF(VVOWT(K3).NE.0) GO TO 261
351             V1=V1+1
352             OWD(V1)=TVLOF(K3)
353     261     V1=V1+1
354             OWD(V1)=2
355
356             RETURN
357     262     IF(K3-10)    ,,290
358             IF(K1.NE.21.OR.K2.NE.9) GO TO 265 @ CONS.+R+YOD
359             OWD(IO)=TVLOF(J)
360             OWD(V1)=18
361             IO=IO+2
362             V1=V1+2
363             OWD(IO)=6
364             CALL MES(OMES(1,IO),7)
365             OWD(V1)=2
366             CALL MES(OMES(1,V1),14)
367             RETURN
368     265     IF(K2-10)    ,,285
369             IF(K1-9) 266,,266
370             IF(J.NE.11.AND.J.NE.12) GO TO 268 @ LABIAL+YOD
371             OWD(IO)=TVLOF(J)
372             OWD(V1)=6
373             CALL MES(OMES(1,V1),7)
374             V1=V1+1
375             OWD(V1)=2
376             CALL MES(OMES(1,V1),14)
377             RETURN
378     266     IF(J-21) 267,,267 @ R+M
379             IF(K1-19) 270,,270
380             OWD(IO)=18
381             IF(K2.EQ.1) OWD(V1)=2
382             GO TO 310
383     268     IF(J.NE.13.OR.J.NE.14) GO TO 269
384             IF(K1.NE.17) GO TO 269
385             OWD(IO)=15
386             IF(K2.EQ.1) OWD(V1)=2
387             GO TO 310
388     267     IF(J-20) 268,,268 @ N+D
389             IF(K1-14) 270,,270
390             OWD(IO)=17
391             IO=IO+1
392             V1=V1+1
393             OWD(IO)=10
394             IF(K2.EQ.1) OWD(V1)=2
395             GO TO 310
396     269     IF(J.NE.13.OR.K1.NE.14) GO TO 270 @ D+T
397             OWD(IO)=10
398             IF(K2.EQ.1) OWD(V1)=2
```

```
399                 GO TO 310
400           270   IF(K1-10    ) 275,275,@ PRIMARY CONS. GP.
401                 IF(K1.EQ.24.AND.K2.EQ.21) GO TO 290
402                 OWD(IO)=TVLOF(J)
403                 OWD(V1)=TVLOF(K1)
404                 V1=V1+1
405                 IF(K1.LE.22.AND.K1.GE.19) OWD(V1)=2
406                 IF(K2.EQ.1) OWD(V1)=2
407                 RETURN
408           275   IF(K1-10)     ,,290
409                 IF(J-23) 280,,280  @ C+ E
410                 IF(K1.NE.2.AND.K1.NE.6) GO TO 285
411                 OWD(IO)=6
412                 OWD(V1)=14
413                 CALL MES(OMES(1,IO),7)
414                 RETURN
415           280   IF(J-9) 285,,285 @ YOD+ I
416                 IF(K1.NE.3.OR .VVOW(2,AT+1).EQ.1) GO TO 290
417                 OWD(IO)=3
418                 GO TO 310
419           285   IF(K1.EQ.1) OWD(V1)=2
420                 IF(VVOW(2,AT+1).EQ.1) CALL MES(OMES(1,V1),2)
421                 DO 290 I1=1,8
422                 IF(J-IOTB3(1,I1)) 290,,290
423                 OWD(IO)=IOTB3(2,I1)
424                 CALL MES(OMES(1,IO),IOTB3(3,I1))
425                 GO TO 310
426           290   CONTINUE
427                 OWD(IO)=TVLOF(J)
428           295   IF(OWD(IO)) ,300
429                 IF(OWD(IO).EQ.6) CALL MES(OMES(1,IO),7)
430                 IF(OWD(IO).EQ.7) CALL MES(OMES(1,IO),8)
431                 IO=IO+1
432           300   IV=IV+1
433                 GO TO 245
434           310   CONTINUE
435                 RETURN
436                 END


@PRT,S      PLAT
```

```
.TH*TPFS.PLAT
    1              SUBROUTINE PLAT
    2              IMPLICIT INTEGER (A-Z)
    3              INCLUDE PARM,LIST
    4              INCLUDE LATIN,LIST
    5              INCLUDE MESAG,LIST
    6              COMMON/PAGE/PG(22,19)
    7              DATA PG/418*' '/
    8              INTEGER TLPLAT(LNCHR)
    9             *          /'A','E','I','O','U','A','E','I','O','U','6','7','8','P',
   10             $'B','T','D','F','V','S','Z','M','N','R','L','C','G','X','H','Q'/
   11              INTEGER XWD(18)
   12              LOGICAL KICK/.TRUE./
   13              CALL NBTW(0,XWD,18)
   14              DO 1 I=1,18
   15              J=LWD(I)
   16              IF(J),2
   17              XWD(I)=FLD(0,6,TLPLAT(J))
   18         1    CONTINUE
   19         2    CALL PCK3(XWD,PG(1,1))
   20              K=3
   21              DO 108 I=1,18
   22              J=XWD(I)
   23              IF(J),110
   24              CALL MOVE(PG(1,K),6,XWD(I),6,1)
   25              DO 100 N=1,3
   26              IF(LMES(N,I)),105
   27              FLD(0,6,PG(2,K))=1R-
   28              I1=LMES(N,I)
   29              CALL MOVE(PG(1,K),9,PMESS(1,I1),1,24)
   30              K=K+1
   31              KICK=.FALSE.
   32        100   CONTINUE
   33        105   IF(KICK) K=K+1
   34              KICK=.TRUE.
   35        108   CONTINUE
   36        110   RETURN
   37              END


.PRT,S     PVUL
```

```
TLTH*TPF$.PVUL
      1              SUBROUTINE PVUL
      2              IMPLICIT INTEGER (A-Z)
      3              INCLUDE PARM,LIST
      4              INCLUDE VULGR,LIST
      5              INCLUDE MESAG,LIST
      6              COMMON/PAGE/PG(22,19)
      7              INTEGER XWD(18)
      8              INTEGER PWD(3)
      9              LOGICAL KICK/.TRUE./
     10              CALL NBTW(' ',PWD,3)
     11              CALL NBTW(0,XWD,18)
     12              DO 1 I=1,18
     13              IF(VWD(I)),1
     14              J=VWD(I)
     15              XWD(I)=FLD(0,6,VCHR(J))
     16        1     CONTINUE
     17              CALL PCK3(XWD,PWD)
     18              CALL MOVE(PG(1,1),34,PWD,1,18)
     19              K=3
     20              DO 108 I=1,18
     21              J=XWD(I)
     22              IF(J),110
     23              CALL MOVE(PG(1,K),39,XWD(I),6,1)
     24              DO 100 N=1,3
     25              IF(VMES(N,I)),105
     26              FLD(18,6,PG(7,K))=1R-
     27              I1=VMES(N,I)
     28              CALL MOVE(PG(1,K),42,PMESS(1,I1),1,24)
     29              K=K+1
     30              KICK=.FALSE.
     31       100    CONTINUE
     32       105    IF(KICK) K=K+1
     33              KICK=.TRUE.
     34       108    CONTINUE
     35       110    RETURN
     36              END


@PRT,S     POFR
```

```
TLTH*TPFS.POFR
      1              SUBROUTINE POFR
      2              IMPLICIT INTEGER (A-Z)
      3              INCLUDE PARM,LIST
      4              INCLUDE MESAG,LIST
      5              INCLUDE OFRNC,LIST
      6              COMMON /PAGE/PG(22,19)
      7              INTEGER PWD(3)
      8              INTEGER XWD(18)
      9              LOGICAL KICK/.TRUE./
     10              CALL NBTW(' ',PWD,3)
     11              CALL NBTW(0,XWD,18)
     12              DO 1 I=1,18
     13              IF(OWD(I)),1
     14              J=OWD(I)
     15              XWD(I)=FLD(0,6,OCHR(J))
     16          1   CONTINUE
     17              CALL PCK3(XWD,PWD)
     18              CALL MOVE(PG(1,1),64,PWD,1,18)
     19              K=3
     20              DO 108 I=1,18
     21              J=XWD(I)
     22              IF(J),110
     23              CALL MOVE(PG(1,K),69,XWD(I),6,1)
     24              DO 100 N=1,3
     25              IF(OMES(N,I)),105
     26              FLD(18,6,PG(12,K))=1R-
     27              I1=OMES(N,I)
     28              CALL MOVE(PG(1,K),72,PMESS(1,I1),1,24)
     29              K=K+1
     30              KICK=.FALSE.
     31          100 CONTINUE
     32          105 IF(KICK) K=K+1
     33              KICK=.TRUE.
     34          108 CONTINUE
     35          110 RETURN
     36              END


@PRT,S     PPG
```

```
TLTH*TPFS.PPG
     1              SUBROUTINE PPG
     2              IMPLICIT INTEGER (A-Z)
     3              INCLUDE PARM,LIST
     4              COMMON/PAGE/PG(22,19)
     5              INTEGER LCTL(3)/'0','20','39'/
     6              DATA LC/1/
     7              INTEGER LCL(2)/'L,   . '/
     8              CALL MOVE(LCL,3,LCTL(LC),1,2)
     9              CALL NPRTCN(PU,LCL)
    10              DO 100 I=1,18
    11       100    CALL NPRINT(PU,PG(1,I))
    12              CALL NBTW(' ',PG(1,1),418)
    13              LC=LC+1
    14              IF(LC.LE.3) RETURN
    15              LC=1
    16              RETURN
    17              END


@PACK       TPFS.


@PREP       TPFS.


@COPY       TPFS.,PHONOFL.
    78 BLOCKS COPIED


@PREP       PHONOFL.


@FIN
```

```
TLTH*TPFS.LTOV
      1              SUBROUTINE LTOV
      2              IMPLICIT INTEGER (A-Z)
      3              INCLUDE PARM,LIST
      4              INCLUDE LATIN,LIST
      5              INCLUDE VULGR,LIST
      6              INTEGER TCLVL(LNCHR)/1,6,2,7,4,1,2,3,4,5,2,6,4,11,12,13,14,15,
      7             $16,17,18,19,20,21,22,23,24,25,26,27/
      8              INTEGER IVCTB1(3,5)/27,9,7,
      9             $14,12,0,
     10             $15,28,9,
     11             $18,28,9,
     12             $19,28,9/
     13              INTEGER IVCTB2(5,7)/27,22,10,19,0,
     14             $14,16,13,13,0,
     15             $15,16,13,13,0,
     16             $24,20,17,17,0,
     17             $26,16,9,13,7,
     18             $14,20,17,17,0,
     19             $14,24,12,21,0/
     20              POS=0
     21              NVV=0
     22              CALL NBTW(0,VWD,RSZ)
     23              CALL NBTW(0,VMES,3*RSZ)
     24              CALL NBTW(0,VVOW,20)
     25              IV=1
     26              IL=2
     27              IF(LWD(1)-29),200  @ H
     28              IF(LWD(1)-27) 100,,100  @G
     29              I=LWD(2)
     30              IF(LPAL(I)),120
     31              VWD(IV)=9  @YOD
     32              CALL MES(VMES(1,IV),7)
     33              IV=2
     34              GO TO 200
     35        100   IF(LWD(1)-20) 110,,110  @S
     36              I=LWD(2)
     37              IF(LVOWT(I)),,120
     38              VWD(1)=6
     39              CALL MES(VMES(1,1),3)
     40              VWD(2)=17
     41              IV=3
     42              GO TO 200
     43        110   IF(LWD(1)-30) 120,,120  @Q
     44              VWD(1)=23
     45              IV=2
     46              IL=3
     47              GO TO 200
     48        120   IF(LWD(1)-16) 130,,130 @D* YOD
     49              IF(LWD(IL)-12) 130,,130
     50              VWD(IV)=9
     51              CALL MES(VMES(1,IV),7)
     52              IV=2
     53              IL=3
     54              GO TO 200
     55        130   IF(LWD(1)-13) 150,,150 @ AU(8)
     56              VWD(IV)=1
```

```
TLTH*TPFS.TLAT
     1              SUBROUTINE TLAT(WD,S)
     2              IMPLICIT INTEGER (A-Z)
     3              INCLUDE PARM,LIST
     4              INCLUDE LATIN,LIST
     5              INTEGER WD(1),WD2(RSZ),NPROC(7)
     6              DATA NPROC/3*' ','-NOT PROCESSED',0/
     7              INTEGER LMT1(LNCHR)/5*0,6*1,0,1,17*0/
     8              NLV=0
     9              CALL NBTW(' ',WD2,RSZ)
    10              CALL NBTW(0,LMES,3*RSZ)
    11              CALL NBTW(0,LVOW,20)
    12              CALL NBTW(0,LWD,RSZ)
    13              CALL SEP3(WD,WD2)
    14              DO 1 I=1,RSZ
    15              IF(WD2(I)),3
    16              DO 2 K=1,LNCHR
    17              IF(WD2(I)-LCHR(K)) 2,,2
    18              LWD(I)=K
    19              CALL MES(LMES(1,I),LMT1(K))
    20              IF(LVOWT(K)) 1,1
    21              NLV=NLV+1
    22              LVOW(1,NLV)=I
    23              GO TO 1
    24        2     CONTINUE
    25              CALL XFER(WD(1),NPROC,3)
    26              CALL NPRINT(6,NPROC)
    27              RETURN 2
    28        1     CONTINUE
    29        3     RETURN
    30              END


@PRT,S      LAC
```

```
TLTH*TPFS.LACC
     1              SUBROUTINE LACC
     2              IMPLICIT INTEGER (A-Z)
     3              INCLUDE PARM,LIST
     4              INCLUDE LATIN,LIST
     5              IF(NLV-1),,1
     6              LVOW(2,1)=1  @ACCENTED
     7              POS=LVOW(1,1)
     8              GO TO 2
     9        1     POS=LVOW(1,NLV-1)
    10              I=LWD(POS)
    11              IF(LVLS(I))3,3
    12              LVOW(2,NLV-1)=1
    13        2     CALL MES(LMES(1,POS),2)
    14              GO TO 5
    15        3     P1=POS+1
    16              P2=POS+2
    17              IF(LWD(P1)*LWD(P2)),4
    18              C1=LWD(P1)
    19              C2=LWD(P2)
    20              IF(LVOWT(C1)+LVOWT(C2)) 4,,4
    21              IF(LSTOP(C1)+LLIQ(C2)-2),4
    22              LVOW(2,NLV-1)=1
    23              GO TO 2
    24        4     IF(NLV-2) 5,6
    25              IF(LVLS(I  )),,5
    26              POS=LVOW(1,NLV-2)
    27              LVOW(2,NLV-2)=1
    28        7     CALL MES(LMES(1,POS),2)
    29        5     RETURN
    30        6     LVOW(2,1)=1       @ ACCENTED
    31              POS=LVOW(1,1)
    32              GO TO 7
    33              END


@PRT,S      PCK3
```

```
TLTH*TPFS.PCK3
     1              SUBROUTINE PCK3(A,B)
     2              IMPLICIT INTEGER (A-Z)
     3              INTEGER A(1),B(1)
     4              CALL NBTW(' ',B,3)
     5              K=1
     6              DO 100 I=1,3
     7              DO 100 J=0,30,6
     8              IF(A(K)),105
     9              FLD(J,6,B(I))=A(K)
    10              K=K+1
    11       100    CONTINUE
    12       105    RETURN
    13              END


@PRT,S     SEP3
```

```
TLTH*TPFS.SEP3
     1              SUBROUTINE SEP3(A,B)
     2              IMPLICIT INTEGER (A-Z)
     3              INTEGER A(1),B(1)
     4              CALL NBTW(' ',B,18)
     5              K=0
     6              DO 100 I=1,3
     7              DO 100 J=0,30,6
     8              IF(FLD(J,6,A(I)).NE.5) GO TO 90
     9              B(K+1)=0
    10              RETURN
    11        90    K=K+1
    12              FLD(0,6,B(K))=FLD(J,6,A(I))
    13       100    CONTINUE
    14              RETURN
    15              END


@PRT,S     MES
```

```
TLTH*TPF$.MES
     1              SUBROUTINE MES(ARY,MN)
     2              INTEGER ARY(1)
     3              IF(MN)1,1
     4              DO 2 I=1,3
     5              IF(ARY(I)-MN),1,
     6              IF(ARY(I)),,2
     7              ARY(I)=MN
     8              GO TO 1
     9        2     CONTINUE
    10        1     RETURN
    11              END


@PRT,S     LTOV
```